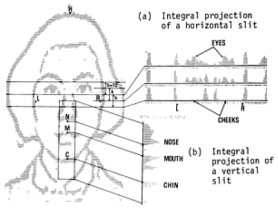
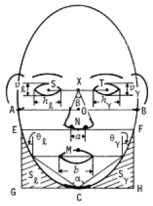
It all began with Takeo Kanade (1973)…

PhD thesis, *Picture Processing System by Computer Complex and Recognition of Human Faces*

- Special purpose algorithms to locate eyes, nose, mouth, boundaries of face
- ~ 40 geometric features, e.g. ratios of distances and angles between features





1

Earlier version used in

Computer Physiognomy

a public attraction at 1970 World Expo

- From talk by Takeo Kanade, *CBMM Face ID Challenge Workshop*

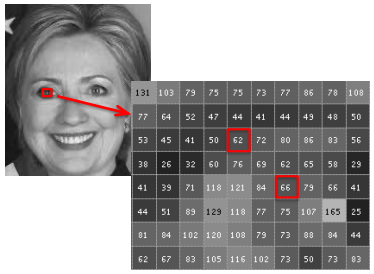
2

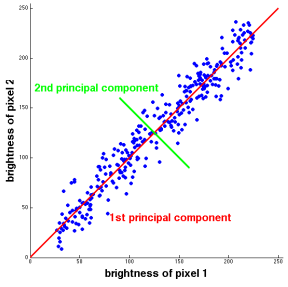
Eigenfaces for recognition (Turk & Pentland)

Principal Components Analysis (PCA)

Goal: reduce the dimensionality of the data while retaining as much information as possible in the original dataset

PCA allows us to compute a linear transformation that maps data from a high dimensional space to a lower dimensional subspace





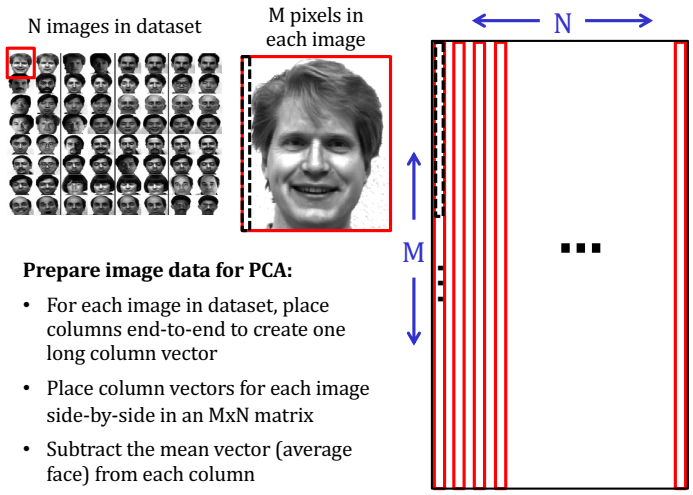
3

Typical sample training set…

Sample images from the Yale face database

- One or more images per person
- Aligned & cropped to common pose, size
- Simple background

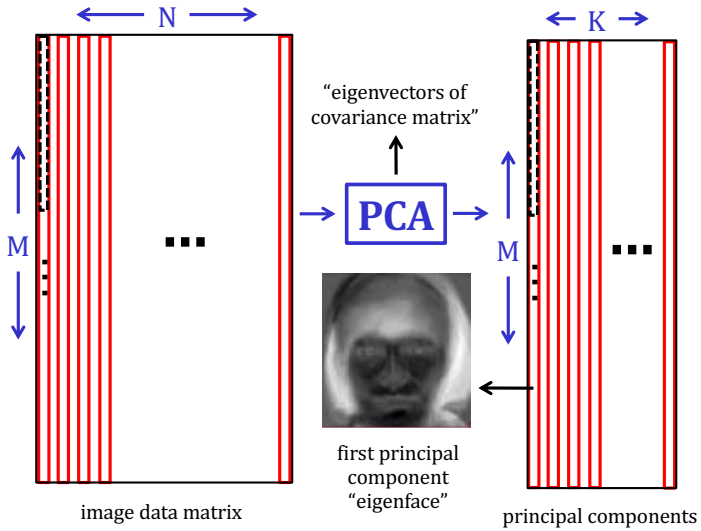
4



Prepare image data for PCA:

- For each image in dataset, place columns end-to-end to create one long column vector
- Place column vectors for each image side-by-side in an MxN matrix
- Subtract the mean vector (average face) from each column

5



6

Eigenfaces for recognition (Turk & Pentland)

Ψ(x,y)

Perform ***PCA*** on a large set of training images, to create a set of *eigenfaces*, $E_i(x,y)$, that span the dataset

First components capture most of the variation across the dataset, later components capture subtle variations

Ψ(x,y): average face (across all faces)

http://vismod.media.mit.edu/vismod/demos/facerec/basic.html

Each face image F(x,y) can be expressed as a weighted combination of the eigenfaces $E_i(x,y)$:

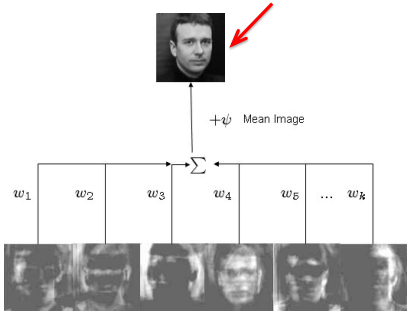
$$F(x,y) = \Psi(x,y) + \Sigma_i w_i * E_i(x,y)$$

7

Representing individual faces

Each face image F(x,y) can be expressed as a weighted combination of the eigenfaces $E_i(x,y)$:

$$F(x,y) = \Psi(x,y) + \Sigma_i w_i * E_i(x,y)$$



Recognition process:

(1) Compute weights w_i for novel face image

(2) Find image m in face database with most similar weights, e.g.

$$\min \sum_{i=1}^{k} (w_i - w_i^m)^2$$

8